



A note on the relative efficiency of methods for computing the transient free-surface Green function

Bingham, Harry B.

Published in:
Ocean Engineering

Link to article, DOI:
[10.1016/j.oceaneng.2016.05.020](https://doi.org/10.1016/j.oceaneng.2016.05.020)

Publication date:
2016

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Bingham, H. B. (2016). A note on the relative efficiency of methods for computing the transient free-surface Green function. *Ocean Engineering*, 120, 15-20. <https://doi.org/10.1016/j.oceaneng.2016.05.020>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A note on the relative efficiency of methods for computing the transient free-surface Green function

Harry B. Bingham

*Dept. of Mechanical Engineering
Technical University of Denmark
2800 Lyngby, Denmark*

Abstract

A number of articles have appeared recently on computing the time-domain, free-surface Green function. Two articles in particular, Chuang et al. (2007); Li et al. (2015) considered the method developed by Clement (1998) who showed that this Green function is the solution to a fourth-order Ordinary Differential Equation (ODE). This ODE has been suggested as a means for speeding up the calculation of the Green function coefficients compared to the standard algorithms developed for example by Newman (1992). Clement solved the ODE using the classical fourth-order, four-step Runge-Kutta scheme (RK44) with a fixed time step size. The two articles mentioned above proposed alternative numerical methods which are claimed to be more efficient. In this note we consider the relative efficiency of these four methods on a representative test case, and conclude that the standard method is the most efficient. Of the ODE-based methods, the method of Chuang et al. (2007) is found to be slightly more efficient than the RK44 method, while the method of Li et al. (2015) is at least an order of magnitude less efficient. It is also pointed out that ODE methods have yet to be extended to include finite water depth.

Keywords: Time-domain Green function, integration of ODEs, wave-structure interaction, marine hydrodynamics

2015 MSC: 00-01, 99-00

URL: www.mek.dtu.dk (Harry B. Bingham)

1. Introduction

The transient free-surface Green function (see Wehausen and Laitone (1960)) is the basis for one class of time-domain panel method codes solving the Neumann-Kelvin linearized seakeeping problem. It is also useful for providing radiation
5 boundary conditions in the far-field of more flexible methods which solve less restrictive versions of the seakeeping problem. By *seakeeping problem* we refer to the interaction between a ship traveling at steady forward speed U and the ocean waves, and we are concerned here with potential flow approximations. By *Neumann-Kelvin* linearization we refer to the linearized version of the problem
10 where the steady disturbance created by the ship is assumed to be a small perturbation to the undisturbed streaming flow. See for example Newman (1977); Ogilvie (1964) for more complete descriptions of this theory. A number of numerical solutions to this problem by means of boundary element methods based on the transient free-surface Green function have appeared over the years, see
15 for example Beck and Liapis (1987); Liapis (1986); King et al. (1988); Korsmeyer (1988); Bingham et al. (1993); Korsmeyer and Bingham (1998); Clement (1998); Kara (2011).

Efficient methods for accurately computing the transient free-surface Green function were initially developed by combining asymptotic series expansions
20 with interpolating polynomials as described for example by Newman (1992). In 1997, Clement discovered that this Green function is the solution to a fourth-order Ordinary Differential Equation (ODE) Clement (1997, 1998). This ODE was envisioned as a means of avoiding the convolution integrals appearing in the time-domain formulation of the problem, and although this goal has not yet been
25 realized, the ODE also provided an alternative means of computing the Green function. Clement adopted the classical explicit four-step, fourth-order Runge-Kutta scheme (RK44) to integrate the ODE and compute the Green function, as did several other subsequent authors including Duan and Dai (2001); Liang et al. (2007). Ten years later, Chuang et al. (2007) developed a closed-form solution

30 to the ODE based on Taylor series expansions, allowing the Green function to
 be computed by truncating the series at a finite number of terms. Although
 not explicitly stated, the implicit intent of this work was to provide a more
 efficient means of solving the ODE compared to the RK44 method, even though
 no demonstration of that result was provided. More recently, Li et al. (2015)
 35 showed that both the Green function and its gradient can be computed from
 the solution to the same ODE. They also applied the Precise Integration (PI)
 method of Liu et al. (2014) to solve the ODE, and showed that this was more
 stable and accurate than RK44 integration when the same time step size is used.
 They also imply that the method is more efficient than the Taylor expansion
 40 method of Chuang et al. (2007), though this is not demonstrated. Comparing
 two numerical methods at the same time-step size is however only relevant when
 the two methods require the same computational effort per time-step, which is
 not the case here. The real question of interest is: which method solves the ODE
 more efficiently? The answer to that question is found by measuring how much
 45 computational time is used by each method to achieve a given level of error.
 The current state of affairs is thus, that no conclusive demonstration has yet
 appeared to support the claim that the use of the Clement ODE can accelerate
 time-domain calculations of wave-body interaction. Neither is it clear which of
 the three currently recommended methods for solving the Clement ODE is the
 50 most efficient.

The goal of this note is to compare the relative efficiency of the four above
 mentioned methods for computing the time-domain free-surface Green function.
 This has been done by implementing all four methods in the same computer
 code, and comparing them on a representative test case where the exact solu-
 55 tion is known. We have attempted to optimize each method so as to minimize
 the computational effort required to achieve a desired level of accuracy. Con-
 firmation that the numerical implementation is reasonably optimal is found by
 recovering the theoretically expected relative scaling of the computational effort
 in each case. From this representative test case, we conclude that the standard
 60 method is the most efficient. Of the three methods for solving the Clement

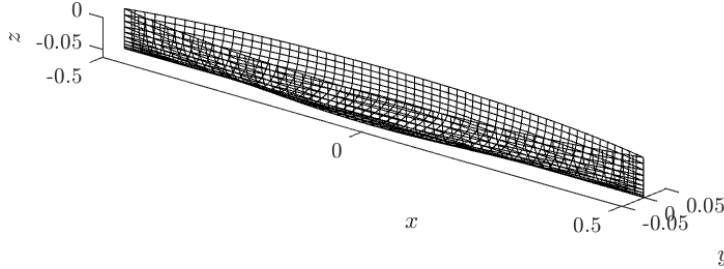


Figure 1: A panel discretization of a ship-like geometry.

ODE, the Taylor expansion method is found to be slightly more efficient than the RK44 method with the PI method at least an order of magnitude less efficient than all of the other methods. We also point out that while the standard method supports finite water depth, this extension has yet to be made for the
65 ODE representation. The promise of the Clement ODE for accelerating time-domain hydrodynamic calculations remains thus enticing but not yet realized.

2. The transient free-surface Green function and the generating ODE

In this Section we review the definition of the Green function and the ODE that can be used to generate it. To establish some context, consider a generic ship represented by a collection of panels as shown in Fig. 1. Only the submerged portion of the geometry is shown and a Cartesian coordinate system is adopted with the z -axis vertical and the x - y plane at the still water level. Consider two points on this geometry $\mathbf{x} = (x, y, z)$ and $\boldsymbol{\xi} = (\xi, \eta, \zeta)$, where the physical variables have been non-dimensionalized by a length scale L (typically the ship length) and the gravitational constant g . In practice, these two points will be collocation points located on two of the panels describing the body surface. In deep water, the velocity potential at the field point \mathbf{x} due to a unit-strength impulsive source at the source point $\boldsymbol{\xi}$ is given by (Wehausen and

Laitone (1960))

$$G(\mathbf{x}; \boldsymbol{\xi}, t) = \delta(t) G_0(\mathbf{x}; \boldsymbol{\xi}) + H(t) \tilde{F}(\mathbf{x}; \boldsymbol{\xi}, t) \quad (1)$$

where $\delta(t)$ is the Dirac delta function and $H(t)$ the Heaviside step function. Here

$$G_0 = \frac{1}{r} - \frac{1}{r'} \quad (2)$$

$$\tilde{F}(R, Z, t) = 2 \int_0^\infty e^{kZ} J_0(kR) \sqrt{k} \sin(\sqrt{k}t) dk \quad (3)$$

70 with

$$R = \sqrt{(x - \xi)^2 + (y - \eta)^2}, \quad r = \sqrt{R^2 + (z - \zeta)^2} \quad (4)$$

$$Z = z + \zeta, \quad r' = \sqrt{R^2 + Z^2}. \quad (5)$$

The wave part of the Green function, $\tilde{F}(R, Z, t)$, can be written as a function of only two variables by making the substitution $kr' \rightarrow \lambda$ to write

$$\tilde{F}(R, Z, t) = \frac{2}{\sqrt{r'^3}} F(\mu, \tau) \quad (6)$$

$$F(\mu, \tau) = \int_0^\infty e^{-\lambda\mu} J_0\left(\lambda\sqrt{1-\mu^2}\right) \sqrt{\lambda} \sin(\sqrt{\lambda}\tau) d\lambda \quad (7)$$

where

$$\mu = -Z/r', \quad \tau = t/\sqrt{r'}. \quad (8)$$

We note that $0 \leq \mu \leq 1$, with $\mu = 0$ being the case where both points lie on the
75 free-surface and $\mu = 1$ the case where the two points lie on the same vertical axis ($R = 0$).

In Clement (1998), it was shown that $F(\mu, \tau)$ is the solution to the following fourth-order ODE

$$F^{(4)} + \mu\tau F^{(3)} + \left(\frac{1}{4}\tau^2 + 4\mu\right) F^{(2)} + \frac{7}{4}\tau F^{(1)} + \frac{9}{4}F = 0 \quad (9)$$

with initial conditions

$$F^{(2k)}(\mu, 0) = 0, \quad F^{(2k+1)}(\mu, 0) = (-1)^k (k+1)! P_{k+1}(\mu), \quad k = 0, 1, \dots \quad (10)$$

80 where P_k are the Legendre polynomials and the notation $F^{(n)} = \partial^n F / \partial \tau^n$ has been adopted. Clement (1998) also derived very similar ODEs for the horizontal and the vertical derivatives of F so that the gradient can be computed by solving two additional ODEs. More recently however, Li et al. (2015) showed that these derivatives can be computed directly from the higher τ derivatives of F .
85 Specifically,

$$\frac{\partial \tilde{F}}{\partial Z} = -\frac{2}{\sqrt{r'^5}} F^{(2)} \quad (11)$$

$$\frac{\partial \tilde{F}}{\partial x} = \frac{x - \xi}{R} F_R \quad (12)$$

$$\frac{\partial \tilde{F}}{\partial y} = \frac{y - \eta}{R} F_R \quad (13)$$

where

$$F_R(\mu, \tau) = \frac{1}{\sqrt{1 - \mu^2}} \left[\frac{3}{2} F + \frac{\tau}{2} F^{(1)} + \mu F^{(2)} \right]. \quad (14)$$

Thus the Green function and its gradient can all be obtained from the solution to Eq. (9). As noted by Dai (2010), this method can be used for non-zero forward speed problems by tabulating the coefficients in the fixed reference frame and
90 then using interpolation with the arguments shifted to the moving frame. A detailed investigation of the accuracy of this procedure has however not yet appeared in the literature (to our knowledge).

3. Numerical solution of the ODE

To compute the Green function from Eq. (9), we evolve forward in time
95 from the initial conditions for a given value of μ out to a predetermined limiting time τ_{lim} . This will naturally be done in a sequence of steps which may or may not be uniformly spaced, but the basic problem is to move the solution from $F(\mu, \tau_0)$ to $F(\mu, \tau_1)$ with a certain level of overall accuracy. The time interval will be denoted by $\Delta\tau = \tau_1 - \tau_0$. We now consider the computational effort
100 required to take one time-step using the methods of interest here.

3.1. Computational effort to take one RK44 time-step

To solve Eq. (9) using the RK44 method, an extended solution vector $\mathbf{Y} = [F, F^{(1)}, F^{(2)}, F^{(3)}]$ is defined, and the fourth-order ODE is converted to a system of four first-order ODEs

$$\mathbf{Y}^{(1)} = \mathbf{A} \mathbf{Y}, \quad (15)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{9}{4} & -\frac{7}{4}\tau & -\left(\frac{\tau^2}{4} + 4\mu\right) & -\mu\tau \end{bmatrix}.$$

To take one time-step requires four evaluations of the right hand side of Eq. (15) or roughly 4^3 FLOPS (Floating Point OperationS), so we will define

$$S_{RK44} = 4^3 \quad (16)$$

105 as the reference computational effort associated with one RK44 time step.

3.2. Computational effort for the Taylor expansion method

The solution to Eq. (9) developed by Chuang et al. (2007) is based on a truncated Taylor series expansion of $F(\mu, \tau)$ (at a fixed value of μ) about the starting time τ_0

$$F(\tau) = \sum_{n=0}^N a_n (\tau - \tau_0)^n \quad (17)$$

where N is the order of the expansion. By inserting Eq. (17) into the ODE and collecting terms at each power of τ , recursive formulae can be derived for the coefficients a_n based on the initial values of \mathbf{Y} . These relations are given by Equations (26) and (27) of Chuang et al. (2007) but it is important to note that both of these equations are missing a minus sign on the right hand side. The computational effort required to evaluate the coefficients a_n , and then compute the solution at the next time-step is proportional to

$$S_{Taylor} = N \log N. \quad (18)$$

110 3.3. Computational effort for the PI method

The PI method applied by Li et al. (2015), first splits the matrix into sub-matrices proportional to $\Delta\tau^n$, $n = 0, 1, 2$ and adds $m + 2$ additional sets of four dummy variables to the solution vector \mathbf{Y} , where $m > 2$. This is essentially a series expansion of the the original solution vector \mathbf{Y} up to order $\Delta\tau^{m+2}$. Now, assuming that the expanded coefficient matrix \mathbf{A} is time-constant, the 2^N algorithm is used to evaluate the matrix exponential $e^{\mathbf{A}\Delta\tau}$ and evolve the solution to the new time-step. The rank of the matrix \mathbf{A} is $4(m + 3)$, and the 2^N algorithm requires a total of $N_2 + 3$ matrix-matrix products where N_2 is the number of iterations applied. The computational effort to take one time-step for this method is thus proportional to

$$S_{PI} = (3 + N_2)(4(m + 3))^3. \quad (19)$$

In Li et al. (2015), an additional parameter H is introduced which represents the number of time-steps over which the matrix \mathbf{A} is held fixed. However, since the computational effort required to reconstruct the matrix \mathbf{A} is negligible compared to the effort required to compute the matrix exponential, there is no
 115 advantage to choosing H to be other than one. We will therefore only consider the case where $H = 1$ and remove this parameter from consideration.

3.4. Confirmation of the scaling estimates

The three methods discussed above have all been implemented into the same Fortran 90 computer program to allow for a timing comparison on one representative platform to verify the scaling estimates shown above. A set of 100 values of μ , evenly distributed between zero and one, and 300 values of $0 \leq \tau \leq 15$ were chosen to test the scaling of the relative computational effort between the methods. Figure 2 shows the ratio of the measured computational effort for the Taylor expansion method to that required by the RK44 method, CPU_{Taylor}/CPU_{RK44} , compared to the predicted value given by

$$C_{Taylor} \frac{S_{Taylor}}{S_{RK44}} \quad (20)$$

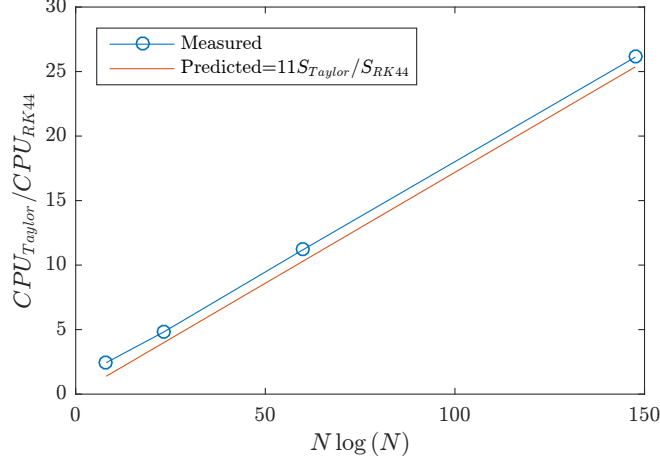


Figure 2: The scaling of the relative computational effort per time-step between the Taylor method and the RK44 method with increasing expansion parameter N .

where the proportionality constant $C_{Taylor} = 11$ has been chosen to give a good fit to the measured data. The values of N plotted here are $N = 5, 10, 20, 40$.

Similarly, Figure 3 shows the ratio of the measured computational effort for the PI method, CPU_{PI}/CPU_{RK44} , compared to the predicted value

$$C_{PI} \frac{S_{PI}}{S_{RK44}} \quad (21)$$

with the constant $C_{PI} = 0.6$ again chosen to give a good fit to the measured data. The (m, N_2) -pairs plotted here are $(m, N_2) = [(5, 5), (10, 10), (20, 15), (40, 20)]$.

The measured ratios can be seen to draw straight lines against the predicted scaling parameters, confirming these estimates. The factors C_{PI} and C_{Taylor} can be expected to vary slightly depending on the platform and the test case parameters chosen, but this example is assumed to be representative.

3.5. Computational effort for the standard method

The TiMIT subroutine which computes the Green function and its gradient using the algorithms of Newman (1992) will be used to represent the standard method TiMIT (1995-1997). There are two steps to the calculation using this

code

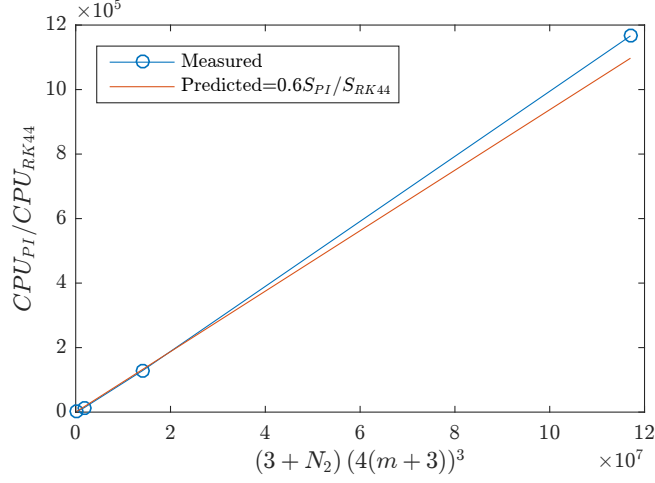


Figure 3: The scaling of the relative computational effort per time-step between the PI method and the RK44 method with increasing parameters m and N_2 .

1. Evaluation of the slowly-varying part of the the Green function based on Chebyshev polynomial expansions. This step requires approximately two hundred FLOPS.
2. Evaluation of the far-field oscillatory part, when necessary. This step requires less than 100 FLOPS.

135

Taking all the required operations into consideration, the RK44 method requires approximately 120 FLOPS per evaluation, so we should expect a ratio of two to three between the two methods. For the test case considered here, the average computational effort for one evaluation using the standard method is found to be very close to twice that required by the RK44 method, so we will adopt this value for the comparisons which are made in the next section,

$$S_{TiMIT} \approx 2S_{RK44}. \quad (22)$$

4. The relative efficiency of the four methods

The relative efficiency of the methods will now be evaluated by considering how much computational time is required to achieve a given accuracy of solution

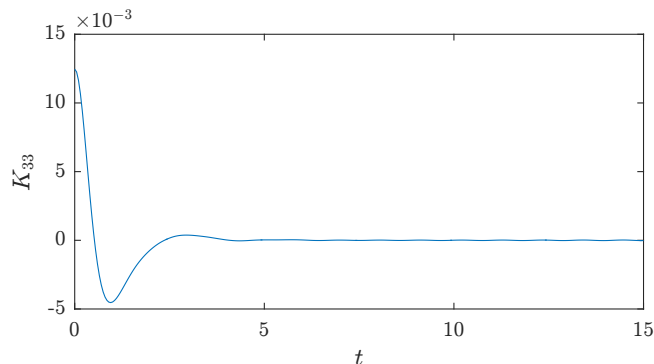


Figure 4: The heave-heave Impulse Response Function (IRF) for the Wigley hull shown in Fig. 1.

at the two limiting cases of $\mu = 0$ and $\mu = 1$ where the exact solution is known.

140 In practice, we are generally interested in pre-computing the Green function and its derivatives at a large number of μ values over a fixed physical time-range, after which the Impulse Response Functions (IRFs) for the body can be computed. For example, for the Wigley hull ship model shown in Figure 1 at zero forward speed, the heave-heave IRF is shown in Figure 4. As can

145 be seen from this typical result, when the length-scale L is chosen to be the largest dimension of the structure, a non-dimensional physical time limit of ten to fifteen is more than adequate. Thus we will define the test case time range to be $0 \leq t \leq 15$ with a time step size of $\Delta t = 0.05$, as this choice typically gives sufficient accuracy when using this type of low-order panel method. When

150 the algorithms developed by Newman (1992) are used to compute the Green function, a relative accuracy of five to six digits is ensured for all values of μ and τ , so this is chosen as the target accuracy level.

We consider now two limiting cases which roughly correspond to the discrete problem of finding the IRFs for the geometry shown in Fig. 1 using a time-

155 domain panel method like TiMIT (1995-1997). For this method, a typical ship geometry will be discretized using approximately 1200 flat panels on which the solution is assumed to be constant. So for this computation, the Green function

needs to be computed at $1200 \times 1200 \times 300 = 432 \times 10^6$ (μ, τ) pairs. Considering all the μ values in this example, at one extreme we have $\mu \approx 0$ and $r' \approx 1$, while
160 at the other we have $\mu \approx 1$ and $r' = .01$, so these values are chosen to define the limiting test cases. Note that because of the relation between the physical time t and the natural time τ (Eq. (8)), it is actually the $\mu = 1$ test case which is more demanding for the ODE methods as it corresponds to $0 \leq \tau \leq 150$ and is therefore more prone to induce instability in the solution. The exact solutions
165 for these two cases are given by

$$F(0, \tau) = \frac{\pi\tau}{2\sqrt{2}} \left[J_{\frac{1}{4}} \left(\frac{\tau^2}{8} \right) J_{-\frac{1}{4}} \left(\frac{\tau^2}{8} \right) + J_{\frac{3}{4}} \left(\frac{\tau^2}{8} \right) J_{-\frac{3}{4}} \left(\frac{\tau^2}{8} \right) \right] \frac{\tau^2}{8} \quad (23)$$

$$F(1, \tau) = \tau e^{-\frac{\tau^2}{4}} M \left(-\frac{1}{2}, \frac{3}{2}, \frac{\tau^2}{4} \right) \quad (24)$$

where J_ν is the Bessel function of order ν and M the confluent hyper-geometric function. We note here that the exact $\mu = 1$ solution is given incorrectly by Li et al. (2015) Eq. (78) with $\frac{3}{4}$ as the second argument of M rather than $\frac{3}{2}$.

	N	m	N_2	$\frac{\Delta t}{\Delta t_{RK44}}$	$\frac{CPU_{TIMIT}}{CPU_{RK44}}$	$\frac{\Delta t}{\Delta t_{Taylor}}$	$\frac{CPU_{Taylor}}{CPU_{RK44}}$	$\frac{CPU_{PI}}{CPU_{RK44}}$
$\mu = 0$:	10	50	20	8	0.25	1	0.5	243,000
$\mu = 1$:	22	50	20	48	$\frac{1}{24}$	5	1.2	42,800

Table 1: Relative effort to compute F at the limiting values of $\mu = 0$ and $\mu = 1$ for $0 \leq t \leq 15$ and $\Delta t = 0.05$ based on the suggested parameters of Chuang et al. (2007); Li et al. (2015).

A time-step size is now established which allows the RK44 method to achieve
170 the desired accuracy level, and this is found to be $\Delta t/8$ and $\Delta t/48$ respectively for the two test cases. This is chosen as the baseline to reflect a strategy of computing the solution to the ODE at a finer resolution in time than might be required to solve the integral equation, and saving only the values that are needed. The PI method remains stable for these two test cases at the physical
175 step size Δt , but to keep the Taylor method stable requires using $\Delta t/5$ for the $\mu = 1$ case. Now, based on the suggested parameter values given by Chuang et al. (2007); Li et al. (2015), we compare the relative efficiency of the four solutions and the results are listed in Table 1. From these results we can see

	N	m	N_2	$\frac{\Delta t}{\Delta t_{RK44}}$	$\frac{CPU_{TIMIT}}{CPU_{RK44}}$	$\frac{\Delta t}{\Delta t_{ay}}$	$\frac{CPU_{Taylor}}{CPU_{RK44}}$	$\frac{CPU_{PI}}{CPU_{RK44}}$
$\mu = 0$:	10	3	4	8	0.25	1	0.5	107
$\mu = 1$:	22	5	5	48	$\frac{1}{24}$	5	1.2	51

Table 2: Relative effort to achieve 5-6 digits of accuracy in the calculation of F at the limiting values of $\mu = 0$ and $\mu = 1$ for $0 \leq t \leq 15$ and $\Delta t = 0.05$ with optimized PI parameters.

that the Taylor method is competitive with RK44, while the PI method is 10^4 -
180 10^5 times slower. The standard method however, is at least twice as fast as all
of the ODE-based methods.

We have attempted to optimize the Taylor and the PI methods by reduc-
ing the suggested parameters, and while we were not successful for the Taylor
method, we find that the PI parameters can be substantially reduced while still
185 giving a relative accuracy of five to six digits. These results appear in Table
2. Even with these choices however, the PI method is still several orders of
magnitude less efficient than the other methods.

4.1. Sensitivity of the comparison to the time limit

The physical time limit set in the above example was based on a typical
190 application of a time-domain method to analyse the wave induced response of
a floating marine structure. In practice other limits might be chosen, so the
sensitivity of the relative efficiencies of the methods to the physical time-span
of the Green function coefficients is investigated here. The same exercise as
described above has been carried out for limiting physical times of half and
195 double the limit adopted above (*i.e.* 7.5 and 30) and the results are tabulated
in Tables 3 & 4. From these results, we can see that changing the time limit

	N	m	N_2	$\frac{\Delta t}{\Delta t_{RK44}}$	$\frac{CPU_{TIMIT}}{CPU_{RK44}}$	$\frac{\Delta t}{\Delta t_{ay}}$	$\frac{CPU_{Taylor}}{CPU_{RK44}}$	$\frac{CPU_{PI}}{CPU_{RK44}}$
$\mu = 0$:	8	3	2	4	0.5	1	0.7	165
$\mu = 1$:	20	3	4	22	$\frac{1}{11}$	3	1.4	40

Table 3: Relative effort to achieve 5-6 digits of accuracy in the calculation of F at the limiting values of $\mu = 0$ and $\mu = 1$ for $0 \leq t \leq 7.5$ and $\Delta t = 0.05$ with optimized PI parameters.

	N	m	N_2	$\frac{\Delta t}{\Delta t_{RK44}}$	$\frac{CPU_{TIMIT}}{CPU_{RK44}}$	$\frac{\Delta t}{\Delta t_{Taylor}}$	$\frac{CPU_{Taylor}}{CPU_{RK44}}$	$\frac{CPU_{PI}}{CPU_{RK44}}$
$\mu = 0$:	12	3	5	18	0.11	1	0.3	58
$\mu = 1$:	40	5	5	90	$\frac{1}{45}$	10	2.8	27.5

Table 4: Relative effort to achieve 5-6 digits of accuracy in the calculation of F at the limiting values of $\mu = 0$ and $\mu = 1$ for $0 \leq t \leq 30$ and $\Delta t = 0.05$ with optimized PI parameters.

changes the details of the comparison but not the conclusions drawn.

It is also important to note that since the asymptotic limit of the Green function at large values of τ is known, see Newman (1992), there is no motivation to numerically integrate the ODE at large time. Any efficient method for computing the Green function will make use of this asymptotic result.

5. Conclusion

Three methods of computing the transient free-surface Green function from the Clement (1998) ODE have been implemented and compared to the standard method based on asymptotic expansions and polynomial interpolation as described for example by Newman (1992). Based on a representative test case, the standard method is found to be more efficient than any of the existing ODE-based methods. Of the three ODE integration methods, the Taylor expansion method of Chuang et al. (2007) is found to be slightly more efficient than the classical RK44 scheme, while the precise integration method of Li et al. (2015) is at least an order of magnitude less efficient than all of the other methods. It is also important to note that while the standard method already supports finite water depth, ODE-based methods have yet to be extended to include this effect. The prospects of speeding up transient hydrodynamic calculations by means of the Clement ODE thus remain enticing, but as of yet, not fully realized.

Acknowledgements

The author would like to acknowledge the ongoing support of the Dept. of Mechanical Engineering, Technical University of Denmark.

References

- 220 Beck, R.F., Liapis, S., 1987. Transient motions of floating bodies at zero speed. J. Ship Research 31, 164–176.
- Bingham, H.B., Korsmeyer, F.T., Newman, J.N., Osborne, G.E., 1993. The simulation of ship motions, in: 6th Intern. Conference on Numerical Ship Hydrodynamics, National Academy of Sciences, USA, Iowa City, Iowa.
- 225 Chuang, J.M., Qiu, W., Peng, H., 2007. On the evaluation of the time-domain Green function. Ocean Engineering 34, 962–969.
- Clement, A., 1997. A shortcut to compute time-domain free-surface potentials avoiding Green function evaluation, in: 12th Intl. Wrkshp. Water Waves and Floating Bodies, <http://www.iwwwfb.org>, Carry-le-Rouet, France.
- 230 Clement, A.H., 1998. An ordinary differential equation for the Green function of time-domain free-surface hydrodynamics. J. Engineering Math. 33, 201–217.
- Dai, Y.Z., 2010. A new algorithm for the time-domain Green function, in: 25th Intl. Wrkshp. Water Waves and Floating Bodies, <http://www.iwwwfb.org>, Harbin, China.
- 235 Duan, W., Dai, Y., 2001. New derivation of ordinary differential equations for transient free-surface Green functions. China Ocean Eng. 4, 499–507.
- Kara, F., 2011. Time domain prediction of added resistance of ships. J. Ship Res. 55, 163–184.
- 240 King, B.W., Beck, R.F., Magee, A.R., 1988. Seakeeping calculations with forward speed using time domain analysis, in: Proc. 17th Symp. Naval Hydro., The Hague, Netherlands.

- Korsmeyer, F.T., 1988. The first- and second-order transient free-surface wave radiation problems. Ph.D. thesis. Massachusetts Institute of Technology. Cambridge, Massachusetts.
- 245 Korsmeyer, F.T., Bingham, H.B., 1998. The forward speed diffraction problem. *Journal of Ship Research* 42:2, 99–112.
- Li, Z.F., Ren, H.L., Tong, X.W., Li, H., 2015. A precise computation method of transient free surface Green function. *Ocean Engineering* 105, 318–326.
- Liang, S., Ren-Chuan, Z., Guo-Ping, M., Yingzhong, L., 2007. A practical
250 numerical method for deep water time domain in Green function. *China J. Hydrodyn.* 3, 380–386.
- Liapis, S.J., 1986. Time domain analysis of ship motions. Ph.D. thesis. The Department of Naval Architecture and Marine Engineering, The University of Michigan. Ann Arbor, Michigan.
- 255 Liu, X.M., Zhou, G., Zhu, S., Wang, Y.H., Sun, W.R., Weng, S.L., 2014. A modified highly precise direct integration method for a class of linear time-varying systems. *China Phys. Mech. & Astron.* 57, 1382–1389.
- Newman, J.N., 1977. *Marine Hydrodynamics*. The MIT Press, Cambridge, Massachusetts.
- 260 Newman, J.N., 1992. The approximation of free-surface Green functions, in: Martin, P.A., Wickham, G.R. (Eds.), *Wave Asymptotics*. Cambridge University Press, pp. 107–135.
- Ogilvie, T.F., 1964. Recent progress toward the understanding and prediction of ship motions, in: *The Fifth Symposium on Naval Hydrodynamics*, Bergen.
265 pp. 3–128.
- TiMIT, 1995-1997. TiMIT; A panel method program for transient wave-body interactions. The Research Laboratory for Electronics, Massachusetts Institute of Technology. Cambridge, MA, U.S.A. <http://www.wamit.com/Publications>.

²⁷⁰ Wehausen, J.V., Laitone, E.V., 1960. Surface waves, in: Handbuch der Physik.
Springer, pp. 446–778.